

AN INVESTIGATIVE STUDY OF MULTISPECTRAL DATA COMPRESSION  
FOR REMOTELY-SENSED IMAGES  
USING VECTOR QUANTIZATION & DIFFERENCE-MAPPED SHIFT-CODING<sup>1</sup>

S. Jaggi, Senior Scientist  
Advanced Sensor Development Laboratory  
Lockheed Engineering & Sciences Company<sup>2</sup>  
Stennis Space Center, MS 39529

N 93 - 22-157  
10/2/82  
P. 16

ABSTRACT

A study is conducted to investigate the effects and advantages of data compression techniques on multispectral imagery data acquired by NASA's airborne scanners at the Stennis Space Center. The first technique used was vector quantization. The vector is defined in the multispectral imagery context as an array of pixels from the same location from each channel. The error obtained in substituting the reconstructed images for the original set is compared for different compression ratios. Also, the eigenvalues of the covariance matrix obtained from the reconstructed data set are compared with the eigenvalues of the original set. The effects of varying the size of the vector codebook on the quality of the compression and on subsequent classification are also presented. The output data from the Vector Quantization algorithm was further compressed by a lossless technique called Difference-mapped Shift-extended Huffman coding. The overall compression for 7 channels of data acquired by the Calibrated Airborne Multispectral Scanner (CAMS), with an RMS error of 15.8 pixels was 195:1 (.041 bpp) and with an RMS error of 3.6 pixels was 18:1 (.447 bpp). The algorithms were implemented in software and interfaced with the help of dedicated image processing boards to an 80386 PC compatible computer. Modules were developed for the task of image compression and image analysis. Also, supporting software to perform image processing for visual display and interpretation of the compressed/classified images was developed.

INTRODUCTION

The exceedingly high data rates of remote sensing instruments have prompted needs for rapid data retrieval, transmission, storage and subsequent processing. These instruments acquire multispectral data for each ground scene element. Thus, several images are created from one spatial scene. These multispectral images stretch the demand on image processing techniques and equipment. Depending on the rate of data acquisition, the volume of data being generated can exceed the available capacities and technologies for data transmission. With continually increasing demands for improved spectral and spatial resolution, the requirements for data handling are likely to become more stringent. Two possible ways to solve this problem are on-board data compression for near-real-time processing and ground-based compression for data archiving. The near-real-time processing could be in the form of reducing the bandwidth of the image with a view towards performing some operation such as unsupervised clustering. This demands that the implementation of the compression scheme be capable of performing fast operations. The implementation of ground-based compression schemes is not as limited as speed is not a constraint.

This work was performed under NASA contract NAS 13-315, Job Order No. 019-CAX01 under the direction of Mr. Bruce Spiering, NASA Project Manager, Advanced Sensor Development Laboratory, at the John C. Stennis Space Center.

Former affiliation. Correspondence should be addressed to: Dr. S. Jaggi, Sverdrup Technology Inc., Engineering and Science Department, Applied Math and Physics Section, John C. Stennis Space Center, MS 39529.

Multispectral data has been tremendously useful in solving problems related to classification of objects through the use of remote sensing techniques. As the sensor systems provide more channels, it becomes increasingly critical to develop and implement methods that reduce the amount of processing without losing any advantages arising from increased amounts of information.

Data compression is a useful tool for the purpose of reducing the bandwidth requirements during data transmission or the memory requirements during data storage. Various techniques have been introduced for this purpose. These techniques can be classified into two broad categories - lossy and lossless. Lossy techniques exploit statistical and spatial correlations in the data to remove the redundancies, thus retaining most of the information. Lossless techniques attempt to perform the same task subject to the constraint of having to perfectly reconstruct the data. Lossy techniques allow for greater compression ratios by reconstructing the data within an error bound. Typical examples of lossy techniques include vector quantization, transform coding, and least means square (LMS) filter. Examples of lossless compression are run-length coding, contour coding and quadtree coding.

This paper describes a study to investigate the compression of multispectral visible and thermal imagery data. The aim is to implement algorithms that perform lossy and lossless data compression on multispectral remotely sensed images. In the case of lossy compression, the objective is to determine the trade-offs associated with the degradation in data which results. In the case of the lossless compression, the degree of compression that is possible is examined.

After the algorithms were implemented on a computer they were tested using a typical set of images obtained by airborne scanners operated and maintained by the Advanced Sensor Development Laboratory at NASA's Stennis Space Center. The scanners are the Thermal Infrared Multispectral Scanner (TIMS) and the Calibrated Airborne Multispectral Scanner (CAMS). The original acquired images were subjected to the algorithms to investigate the amount of compression that is possible. This amount of compression depends on the nature of the data and the number of channels of the imagery. The objectives were different for lossless and lossy techniques.

In the case of lossless compression, only the compression ratio possible needed to be determined. In the case of lossy techniques, the ratio was programmable. The higher the compression ratio, the greater is the degradation in the reconstructed images. This study addresses the problem by studying the amount of degradation that results versus the compression ratio.

Various criteria were used for studying this degradation. One criterion is the difference between the original and the reconstructed sets of images. The closer the two sets of images are, the lesser is the degradation. The degradation is represented as the RMS error between the original and the reconstructed images. Another criterion is to compare the eigenvalues of their covariance matrices. This reveals the amount of correlation fidelity that is lost due to the compression.

Since lossless data compression can always be done after the lossy technique, it is available as an enhancement to the compression performance. However if the data needs to be interpreted in its compressed form, it may not be possible to do so after lossless compression. This is because lossless techniques usually store the data in a form that has no immediate perceptible relation to the original data. This issue is also addressed in this work, where the technique addressed will not only compress the data, but leave the result in a form that will allow the investigator to visually analyze the data. In short, it will not only compress but also give the added benefit of classifying the data in an unsupervised manner.

## LOSSY DATA COMPRESSION

### Background and Introduction

Several techniques have been introduced that reduce the spectral dimensionality of the data, thereby reducing the problem to a computationally manageable one. The spectral dimensionality of the data is defined as the number of channels of the imagery being used for analysis. Most of these techniques are modified forms of the Karhunen-Loeve Transform or the Principal Components Analysis, which by itself is computationally inefficient to implement. This transform exploits the band-to-band correlation of various regions of the imagery. Using these correlations, the redundancy in the multispectral set is eliminated. A classified image is the result of the process.

A singular value decomposition of the image yields the eigenvalues and eigenvectors of the image. The number of eigenvalues equals the number of channels in the sensor data. The total variance of the image set is the sum

of all the eigenvalues of the covariance matrix. However, it is observed that for most images a significant portion of the total variance in the multispectral images is contained within a few dominant eigenvalues. Thus, the eigenvector images corresponding to these dominant eigenvalues are said to adequately represent the multispectral imagery with a minimum loss in total image variance.

The disadvantage with this method is that it is not yet possible to implement it real time. Also, the resultant eigenvector images are transformed versions of the earlier multispectral raw data. Thus, by themselves, they do not offer any clear interpretation of the imagery. Typical methods to perform multispectral classification are the maximum likelihood (ML) and the Principal Components (PC) Analysis. The ML method requires a priori knowledge of the behavior of the statistics of the data. Based upon those statistics, various regions in the images are classified. The ML method belongs in the category of supervised classification techniques. The PC method decomposes the set of multispectral images into an equivalent set of orthogonal images, with each image being an eigenvector of the original set. Using the Karhunen-Loeve transform, the three eigenvector images corresponding to the three highest eigenvalues can be coded in a RGB format, to display a classification of the dominant correlations in the data. The PC method belongs in the category of unsupervised classification techniques.

In this section, we investigate a technique to be used for on-board data-compression of multispectral images obtained from airborne scanners. The airborne scanners of interest in this study are the TIMS and the CAMS which are NASA instruments operated by Sverdrup Technology at the Stennis Space Center. The TIMS is a thermal infrared sensor with six channels in the 8-12 micron region of the electromagnetic spectrum. The CAMS is a visible, near-IR and thermal sensor with eight channels in the visible and near-IR and one broad-band thermal channel. This paper only discusses the results of the CAMS imagery.

Here a method based on the vector quantization of the multispectral images is investigated for its use as a tool for unsupervised image clustering and subsequent data compression. The computational simplicity and the ease of implementation after the quantization parameters have been determined make vector quantization an attractive tool for data processing. The vector is defined in the multispectral imagery context as an array of pixels from the same location from each channel. The rate of compression is programmable depending on the size of the codebook. However, the higher the compression ratio, the greater is the degradation between the original and the reconstructed images. The compressed images are reconstructed and compared with the original set using the PC method. The eigenvalues of the covariance matrix obtained from the reconstructed data set are compared with the eigenvalues of the original set. This comparison is done for varying degrees of compression which are obtained by varying the size of the vector codebook. Also, the error obtained in substituting the reconstructed images for the original set is compared for different compression ratios. The effects of varying the size of the vector codebook on the quality of the compression and on subsequent classification are also presented. The eigenvalues of the covariance matrix of the original multispectral data-set were found to be highly correlated with those of the reconstructed data-set.

The section on lossy compression is organized into four parts. An introduction to vector quantization is followed by a description of the algorithm used to determine the optimal codebook. This is followed by a description of the imagery used for data analysis and the results. In conclusion a comparison of this technique is made with the results obtained from the PC technique.

### Vector Quantization

Vector quantization is a scheme for mapping a large set of vectors into a much smaller set of vectors called codewords. In the case of multispectral data a vector is defined as an array of the pixel elements corresponding to a given location for all the channels of the imagery.

Goldberg et. al. , Gray, and Nasrabadi et. al. provide comprehensive details about the concepts. Hang et. al. and Ramamurthi et. al. discuss three variations on the algorithm.

Vector Quantization is defined as a mapping  $Q$  of  $k$ -dimensional Euclidean space  $R^k$  into  $Y$ , a finite subset of  $R^k$ .

$$R^k \xrightarrow{Q} y \quad y \in Y$$

It can also be interpreted as an encoder-decoder operation. The encoder views an input vector  $x$  and generates the address or the code of the reproduction vector  $Q(x)$ .

$$Q(x) = y \quad \{ x \in R^k : y \in Y \}$$

For multispectral data,  $k$  is the number of channels in the sensor. The decoder uses this address to reconstruct the reproduction vector  $y$ . The set of all  $y$ 's,  $Y$  which is made up of all the constituent codes, is called the codebook and its elements i.e  $y$  are called the codewords or reproduction vectors. The criteria of choosing the codewords is to minimize the cost or penalty associated with reproducing the vectors  $y$  from  $x$ .

$$\text{Cost Function} = \| x - y_{cl} \|^2$$

where  $y_{cl}$  is a vector chosen from the codebook  $y$  such that it is closest to the vector  $x$ .

Thus for a 6 channel ( $k = 6$ ) set of 512 X 512 pixel images, there exist  $512 \times 512 = 262,656$  vectors, each of length 6 elements. This entire set of vectors is partitioned into a finite collection of subsets  $N$ . A codeword is chosen as a representative vector for each subset. A collection of these codewords is called a codebook. The codebook is assumed to be representative of the entire subset, such that any vector in the subset can be represented by one of the codewords in the codebook, with a minimum error between the original vector and its reproduction vector. Figure 1 shows the block diagram of a VQ implementation. An input vector is quantized by choosing the closest codeword from the codebook. Once the codebook has been determined, the encoder maps each input vector to one of the codewords in the codebook. For the purpose of storage or transmission, the only necessary information is the index of the codeword in the image and the codebook. For the purpose of regenerating the images, the decoder compares the index of each pixel location with the corresponding vector in the codebook. By choosing the codeword corresponding to the transmitted index, the output vector is created.

It has been shown by Shannon, that a vector quantizer will always give a better noise performance than a scalar quantizer. Among the advantages of VQ is its easy implementation. Once the codebook has been determined it can be easily developed into a hardware implementation that just performs a look-up operation. Also, if a large training set is used to determine the codebook, this universal codebook can be representative of a wide variety of features in the imagery. This could be especially useful for terrestrial data which may not exhibit many unique features. Figure 2 shows the basic concept involved in vector quantization for multispectral imagery. The vector, which is the array of the pixel values of all channels for a location, is quantized and the resulting codeword is substituted for the original vector. Thus the entire reconstructed set is made up of a finite set of vectors that make up the codebook.

The compression ratio is dependent on the number of codewords needed to represent the multispectral image. Since the vectors are being created in a multispectral manner, only one image needs to be stored. This image contains a spatial distribution of the codewords that represent the multispectral set. However, instead of storing the codewords, the index of the codewords is adequate as a symbol of the codeword. The number of bytes occupied by the original multispectral set (which contains 1 byte per pixel) is given by

$$\text{Original set size} = \text{No. of channels} * \text{No. of rows} * \text{No. of columns.}$$

After the image is compressed only the index representing the spatial distribution of the codewords needs to be stored. The index can be the binary address of the codewords. Thus if there are  $N$  codewords, the number of bits needed to address them is  $\log_2(N)$ . This leads to the expression of the compression ratio using vector quantization for multispectral imagery data

$$\text{Compression Ratio} = \frac{\left( \frac{\text{No. of Channels}}{\log_2(\text{No. of Code words})} \times \text{No. of bits per origl. pixel} \right)}{1}$$

The number of bits per original pixel is 8. The compression ratio is also expressed as a bit-rate. This is the average number of bits required to represent the compressed image. The expression for the bit-rate is given as

$$\text{bits per pixel (bpp)} = \left( \frac{\log_2(\text{Number of code words})}{\text{Number of channels}} \right)$$

A discussion of the algorithm used to obtain these codewords follows.

### Algorithm

The main problem associated with VQ is the design of an optimal codebook so as to minimize the cost of substituting the original vectors with the reproduction vectors  $y$ . This cost measure is denoted as  $d(x, y)$ . A common measure used to estimate this distortion is the Euclidean distance between the original vector and its reproduction. Fig. 3 shows the steps involved in performing vector quantization. The codebook is initialized with a set of vectors that are selected from the original vector set. The entire vector set is then classified into these codewords. Using an algorithm, these codewords are then modified to minimize the distortion between the original set and the reconstructed set. This procedure is repeated until the codewords are optimized.

A set of training vectors can be used to perform the minimization. The codebook thus obtained is optimal in the mean square error sense. Several algorithms have been proposed to obtain this codebook. The most widely used algorithm is commonly known as the Linde-Buzo-Gray (LBG) algorithm.

In the LBG algorithm, an initial reproduction vector set, i.e the codebook  $C[0]$ , is chosen. The criteria for choosing this initial set can vary as well. In this study, it was decided to space the initial vectors a minimum distance apart. Vectors were picked up from the image at random until the required number,  $N$  were selected. A threshold is assigned as the minimum acceptable cost or penalty between the original vectors in the training set  $x$  and the reproduction vectors  $y$ . When the average error due to the quantization of the vectors is equal to or below this threshold, the codebook is said to be optimal.

Each vector in the training set is quantized using the initial codebook  $C[0]$ . For every vector in the training set  $x$ , a vector  $y$  from the codebook that is closest to  $x$ , is chosen. The average error due to the quantization of  $x$  to  $y$  is computed. The number of vectors  $x$  is determined by the size of the training set. The number of vectors in the codebook  $N$ , is fixed depending upon the level of quantization required. Each codeword in the existing codebook is replaced by the centroid of all the training vectors assigned to the codeword. This process is repeated until the error goes below a prespecified threshold.

### Imagery Data Analysis

Before beginning the analysis of the images, a criterion needs to be established for comparing the original and the reconstructed images. Two criteria were determined to be of critical importance.

In remote sensing, it is important to be able to determine the correlations between the various multispectral images obtained. These correlations reveal various facts about the data being mapped. Thus it is important that if the original data is being transformed (i.e being subjected to lossy compression), its statistical correlations be maintained as closely as possible. This leads to an analysis of the principal components of the original and the reconstructed images. The eigenvalues of the images were obtained by singular value decomposition. The total variance of the multispectral image-set is the sum of the diagonals of the covariance matrix.

It is also important that the reconstructed images be as similar to the original images as possible. The measure chosen is the root mean square difference between the original and the reconstructed images.

$$\epsilon_{\text{RMS}} = \sqrt{\frac{\sum_{\forall \text{pixels}} \|x - y_{cl}\|^2}{\text{No. of pixels} \times \text{No. of channels}}}$$

The data used for the analysis was obtained by airborne scanners operated and maintained by the Advanced

Sensor Development Laboratory at the Stennis Space Center for NASA. Figure 4 details the characteristics of the scanners. The CAMS data were acquired over western Puerto Rico in January 1990 over land and water. The aim was to study impacts of man-induced changes on land that affect sedimentation into the near-shore environment.

The algorithm was coded into a PC-based system used for the purpose of image processing. This system uses two plug-in boards for performing basic image processing operations. The first board DT-2861 is a frame grabber that also has a 16 image buffer for performing simultaneous on-board imaging operations without having to access the hard disk for different sections of the image. The other board DT-2858 is used to accelerate image processing operations by having them done in hardware.

### Results of Vector Quantization

A channel of the original image-set is shown in Fig. 5 for CAMS. The codebooks for different codewords were obtained after 15 iterations. The components of the two sets of the data were then compared to check for the accuracy of the vector quantization process. Figure 6 shows the RMS error for different compression ratios for CAMS. Figure 7 shows the total variance of the reconstructed set. The total variance is the sum of all the eigenvalues of the covariance matrix. Figure 8 shows the relative variance of each eigenvalue. The numerical values of these relative eigenvalue distributions are shown in tabular form in Fig.9a. Figures 9b-d show the percentages for the first, second and the third most dominant eigenvalues of the covariance matrix of the reconstructed data-set. From the two sets of matrices it is obvious that most of the information is contained in the two dominant eigenvalues for which there is significant correlation between the original and the reconstructed data set as the number of codewords is increased. Figures 10-11 show the corresponding results for the CAMS data-set. The images show the indexes for the codewords and are not related to the original images. Each index corresponds to a vector which can be used to generate the reproduced data-set. The nature of complexity in Fig. 11 corresponding to 128 codewords can be contrasted with that of 4 codewords in Fig. 10.

### Hardware Implementation

The implementation of the lossy data compression technique involves two stages. In the first stage the codebook is determined. This is done using the algorithm as described earlier. The algorithm requires as its input the compression ratio desired and the multispectral image data set. Its output is the optimized set of codewords that represent the imagery set. Each codeword is a vector with its number of elements equaling the number of channels in the multispectral set. In the second stage, each pixel location in the multispectral set is replaced with the codeword that is closest to it. This is the process of quantization. The original vector made up of the value of the pixel at a specific location in all channels is replaced by another vector which is picked from the codebook such that, of all the vectors in the codebook, this new vector is closest to the original vector. Thus the original vector is quantized to the new vector. Since the codebook has a fixed number of vectors or codewords, each codeword in the codebook is given an index number. For example a codebook with 16 vectors can have indexes from 0 through 15. This index of the new vector is sufficient to represent the new vector. Thus, each vector in the original image is replaced by the index of the codeword in the codebook that is closest to it. The information as to which index is what vector, i.e the codebook, can be tagged along at a later stage.

Figure 12 shows a possible configuration of the implementation. Once the codebook has been determined, the operation is reduced to almost a look-up-table approach. The incoming vector is compared with the codewords in the codebook and the index of the codeword closest to the vector is transmitted. If each index is denoted by a color, this information can also be displayed on a real-time monitor. There is hardware available to perform these operations. The only problem that still remains is that of finding the optimum codebook in a near-real-time manner. This can be achieved by a parallel scheme. While the existing codebook in the board is being used to compress the data, another codebook is in the making in the background. This can be done by an independent processing unit, which is dedicated to implementing the LBG algorithm. After it has optimized on a codebook it then updates the codebook currently in use, with the new codebook. However, every time a new codebook is used it needs to be transmitted. All subsequent indexing will be in reference to the new codebook. A criterion could be introduced for the update. The existing codebook is replaced only if the difference between the existing and the new codebook exceeds a certain threshold. Since terrestrial data is low frequency in nature and does not change very rapidly this threshold will obviate the need to constantly update the codebook and transmit a new set of codewords with the data. Also, if the ground scene does indeed change, this will change the codebook significantly and the threshold will allow for the replacement of the codebook.

## Conclusion

The vector quantization technique is an effective tool for data compression and classification. Amongst the advantages of this technique is that it is easy to implement. It also effectively exploits the redundancy present in the channel-to-channel correlations of the data to reduce the memory required for the storage of the images.

## LOSSLESS COMPRESSION

### Introduction

In this section a technique for coding the same remotely sensed data with 100% restoration is investigated. This technique involves difference mapping and shift-extended coding of the original data. The first step is mapping of the data with the use of a difference transform. This mapped data is then used to generate a set of symbols. These symbols are then coded through a Huffman coder. Figure 13 shows the high-level description of these main functions.

### Mapping Transform

The lossless coding technique discussed here can be divided into two parts - a mapping function and a symbol generating function. The mapping function used for this study is a difference mapping. Each pixel is mapped as the difference between the present and the previous pixel. Denoting the 'i'th pixel by  $x[i]$  and the 'i'th map by  $m[i]$ ,

$$m[i] = x[i] - x[i-1]$$

The advantage of this mapping is that the data, which is not changing rapidly on a pixel by pixel basis, can be condensed to a smaller dynamic range. For an 8 bit system, the pixel has a range of 0 - 255. The mapped function theoretically has a range of -255 -> +255. Assuming that the pixels are not changing rapidly, most of the values of the mapped function can lie within a much narrow range ( say -4 -> +4 ). This number is fixed arbitrarily based on the statistics of the data. For reasons stated later in the section, a size that is a power of 2 is chosen.

$$N = 2^{**b}$$

where  $b$  is an integer. It is called the bit-size of the code.

The aim is to code the entire mapped set into a series of symbols that can be easily interpreted. The symbols available for the coding are 0, 1, 2, ...  $N-2$ ,  $N-1$ . However the output of the difference transform could also be negative, e.g., -4 -> +4. Hence the problem still remains as to the representation of negative numbers. The symbols are thus made to represent numbers that fall within the range:  $-N/2$ ,  $(-N/2 + 1)$ ,  $(-N/2 + 2)$ , ... 0, 1, 2, ...,  $(N/2 - 2)$ ,  $(N/2 - 1)$ . Also, there are numbers that are not going to fall within this range. This problem is solved by using extender symbols. Thus the symbol 0 is not used to denote  $-N/2$ . Instead, it is used to denote that the mapped number  $m[i]$  is less than or equal to  $-N/2$ . Similarly, the symbol  $N-1$  is not used to denote  $(N/2-1)$ . Instead it is used to denote that  $m[i]$  is greater then or equal to  $(N/2 - 1)$ . The other symbols retain their previous assignments. Thus the symbols 0 and  $N-1$  are shift extender symbols. The entire mapped function can now be coded into these series of symbols. Each of these symbols now has a value between 0 and  $N-1$ . Fig. 14 describes the detail operation.

### Huffman Coding

If the symbols themselves were used for the codes, the result would be a fixed-length compression, i.e., a word 'b' bits in length could be used to represent every symbol. However all the  $N$  symbols may not be uniformly distributed over the data. It is possible that some of the symbols have a much higher occurrence than the other symbols. This fact could be exploited by using a Huffman code on the data. This code assigns a bit-length to each symbol that is inversely related to the probability of occurrence of the symbol in the data. Thus symbols that occur more often are assigned a shorter bit-length than the symbols that occur less frequently. This results in further compression. The Huffman code requires that  $N$  symbols with a bit-length of  $b$  bits (such that  $N = 2^{**b}$ ) be fed into the algorithm. This is the reason for choosing the number  $N$  earlier, such that it is a power of

2. An optimum assignment is done only if there is a finite probability that can be associated with each of the N possible symbols.

### Results of Lossless Compression

The image that resulted from the lossy compression was further compressed by using this lossless technique. The results of the analysis are shown in Figs. 15a-b. The greater the bit-size, the higher the compression possible. However, the complexity of generating the code increases, thereby increasing the computation time. Usually, the level at which the rate evens out is an acceptable bit-size. The compression ratios are higher for images that had fewer numbers of codewords. However, as was discussed in the section on lossy compression, there is a trade-off between information lost and compression attained.

## OVERALL COMPRESSION RESULTS

The overall compression is achieved by multiplying the compression ratios achieved by the lossy and the lossless algorithms. Figure 16 shows the results of compression on the CAMS data-set. The overall compression possible for the 7 channel CAMS data with an RMS error of 15.8 pixels was 195:1 and with an RMS error of 3.6 pixels was 17.8:1.

## SUMMARY

A feasibility study was conducted to investigate the advantages of data compression techniques on multispectral imagery data. Two different techniques were implemented on remotely sensed data acquired from airborne scanners maintained and operated by NASA at the Stennis Space Center.

The first technique called **Vector Quantization** was used for lossy compression. The vector is defined in the multispectral imagery context as an array of pixels from the same location from each channel. The total number of original vectors is equal to the size of the image. Each of these vectors is quantized to a set of optimum vectors. This set is called the codebook. Since the size of the codebook is much smaller than the total number of original vectors, significant compression results. The rate of compression is programmable. However the higher the compression ratio, the greater is the degradation between the original and the reconstructed images. The analysis for 6 channels of data acquired by the Thermal Infrared Multispectral Scanner (TIMS) resulted in compression ratios varying from 24:1 (RMS error of 8.8 pixels) to 7:1 (RMS error of 1.9 pixels). The analysis for 7 channels of data acquired by the Calibrated Airborne Multispectral Scanner (CAMS) resulted in compression ratios varying from 28:1 (RMS error of 15.2 pixels) to 8:1 (RMS error of 3.6 pixels). The technique of Vector Quantization can also be used to interpret the main features in the image, since those features are the ones that make up the codebook. Hence, Vector Quantization not only compresses the data, but also classifies it.

The compressed images are then reconstructed and compared with the original set using the Karhunen-Loeve Transform through the Principal Components Analysis. The eigenvalues of the covariance matrix obtained from the reconstructed data set are compared with the eigenvalues of the original set. This comparison is done for varying degrees of compression which are obtained by varying the size of the vector codebook. Also, the error obtained in substituting the reconstructed images for the original set is compared for different compression ratios. The effects of varying the size of the vector codebook on the quality of the compression and on subsequent classification are also presented. The eigenvalues of the covariance matrix of the original multispectral data-set were found to be highly correlated with those of the reconstructed data-set.

The second technique, called **Difference-mapped Shift-extended Huffman coding**, was 100% lossless i.e it resulted in images that were capable of complete restoration. Initially, the data was mapped to a difference transform. This transformed image was then converted into symbols using a specific bit-size. These symbols were then coded using Huffman coding. The output data from the Vector Quantization algorithm was further compressed without any increase in the RMS error by subjecting it to this technique. The TIMS data resulted in additional compression of 5.33 (for 24:1 compressed image) to 1.28 (for 7:1 compressed image). The CAMS data resulted in additional compression of 7 (for 28:1 compressed images) to 2.22 (for 8:1 compressed images).

Thus, the overall compression possible for the 6 channel TIMS data with an RMS error of 8.8 pixels was 128:1 and with an RMS error of 1.98 pixels was 8.8:1. The overall compression possible for the 7

channel CAMS data with an RMS error of 15.8 pixels was 195:1 and with an RMS error of 3.6 pixels was 17.8:1.

The algorithms were implemented in software and interfaced with the help of dedicated image processing boards to an 80386 PC compatible computer. Modules were developed for the task of image compression and image analysis. These modules are very general in nature and are thus capable of analyzing any sets or types of images or voluminous data sets. Also, supporting software to perform image processing for visual display and interpretation of the compressed/classified images was developed.

## REFERENCES

- Goldberg, M. and Sun, H.F., 1986. Image Sequence Coding Using Vector Quantization. *IEEE Transactions on Communication*, v. Com-34, iss. pp. 703-710.
- Gonzalez, R. and Wintz, P., 1987. Digital Image Processing. Addison-Wesley Inc. Reading, MA, pp. 255-330.
- Gray, R. M., 1984. Vector Quantization. *IEEE ASSP Magazine* pp. 4-29.
- Hang, H.M., and Haskell, B.G., 1988. Interpolative Vector Quantization Of Color Images. *IEEE Transactions on Communication*, v. Com-36, iss. 4, pp. 465-470.
- Hang, H.M., and Woods J.W., 1985. Predictive Vector Quantization Of Images. *IEEE Transactions on Communication*, v. Com-33, iss. 11, pp. 1208-1219.
- Huffman, D. A., 1952. A method for the construction of minimum redundancy codes. *Proceedings IRE* v. 40, pp. 1098-1101.
- Leger, A., Omachi, T., Wallace, G. K., 1991. JPEG still picture compression algorithm. *Optical Engineering*, v. 30, no. 7, pp. 947-954.
- Linde, Y., et al, Buzo, A., 1980. An Algorithm For Vector Quantizer Design. *IEEE Transactions on Communication*, v. Com-28, iss. 1, pp. 84-89.
- Nasrabadi, N.M., and King, R.A., 1988. Image Coding Using Vector Quantization: A Review. *IEEE Transactions on Communication*, v. 36, iss. 8, pp. 957-971.
- Panchanathan, S., Goldberg, M., 1991. A content-addressable memory architecture for Image Coding using Vector Quantization. *IEEE Signal Processing*, v. 39, no. 9, pp. 2066-2078.
- Ramamurthi, B., and Gersho, A., 1986. Classified Vector Quantization of Images. *IEEE Transactions on Communication*, v. Com-34, pp. 1105-1115.
- Shannon, C.E., 1948. A Mathematical Theory Of Communication. *Bell Systems Technical Journal* 27, pp. 379-423, 623-656.

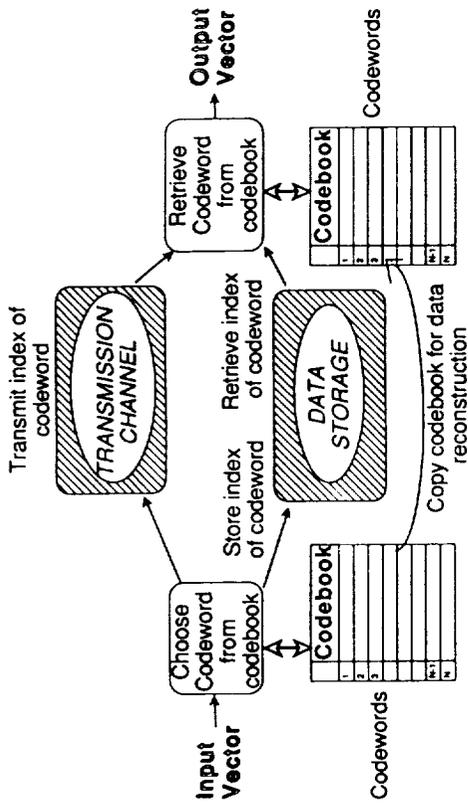


Fig. 1 Vector Quantization - the concept.

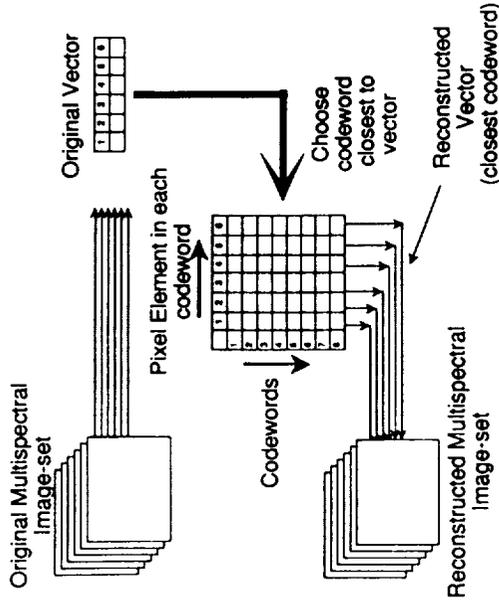
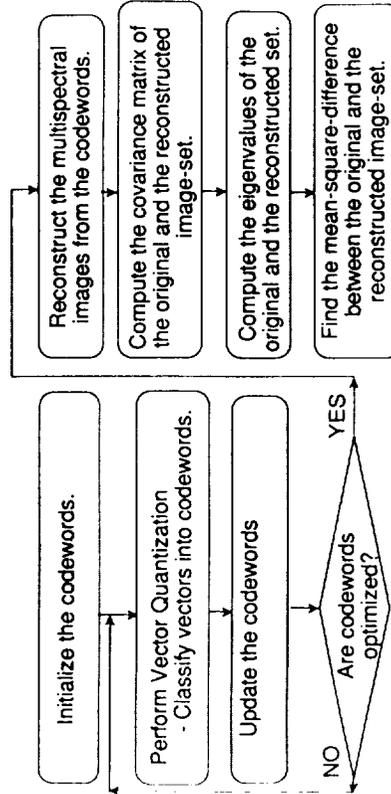


Fig. 2 Spectral Vector Quantization



FIND CODEWORDS

ANALYZE IMAGES

Fig. 3 VECTOR QUANTIZATION - procedure & analysis.

ATTRIBUTES	TIMS	CAMS
Type of sensor	Airborne	Airborne
Type of radiation	Thermal infrared	Visible & infrared
Number of channels	6	9
Thermal bands	6 (8.2 - 12.2)	1 (8 - 12)
Visible bands	0	6 (.4 - .7)
Near Infrared bands	0	2 (.9 - 2.3)

Fig. 4 NASA Airborne Scanners at SSC.



Fig. 5. CAMS acquired data - Channel 3.

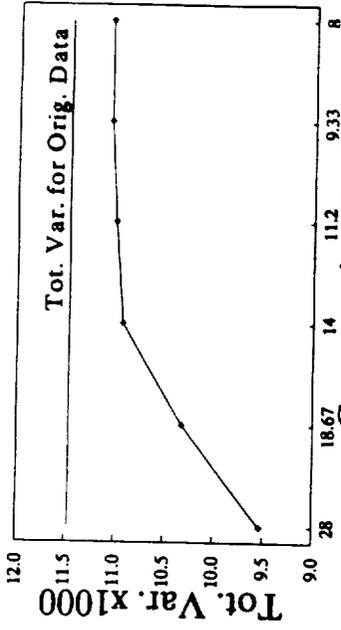


Fig. 7 Total Variance of Reconstructed Images

	Eigenvalues							
	1	2	3	4	5	6	7	SUM
128	8.00	.95134	.04472	.00321	.00059	.00001	.000037	.000056
64	9.33	.90651	.08800	.00463	.00057	.00020	.000027	.000027
32	11.20	.87112	.11604	.01030	.00202	.00036	.000065	.000065
16	14.00	.86408	.12271	.01070	.00221	.00013	.000079	.000079
8	18.67	.86178	.12382	.01140	.00245	.00018	.000188	.000188
4	28.00	.86033	.12429	.01242	.00258	.00026	.000051	.000051
ORIGINAL	1	.97754	.01617	.00481	.00099	.00028	.000190	.000170

Fig. 9a Relative distribution of variance in multispectral CAMS Imagery data.

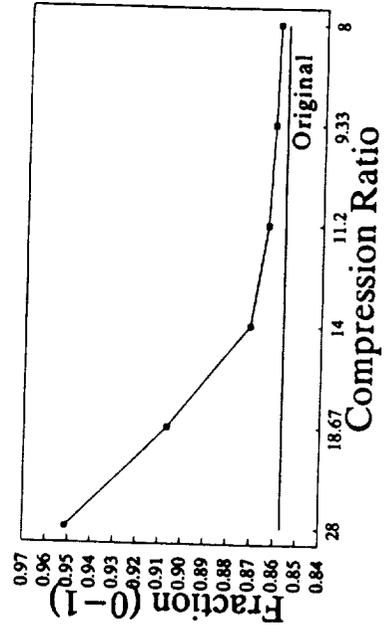


Fig. 9b Fraction of Total Variance in Eigenvalue No. 1

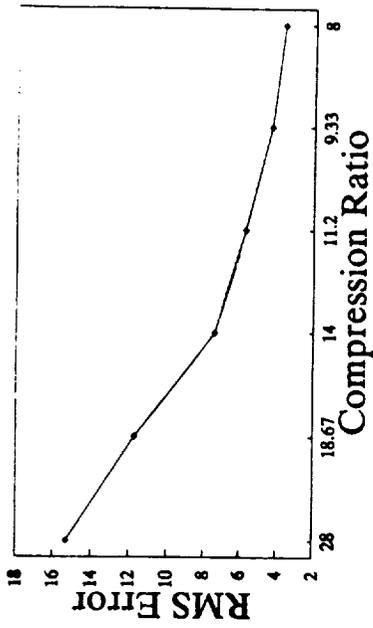
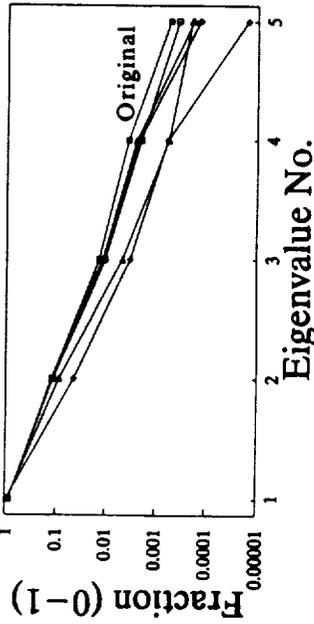


Fig. 6 RMS Error Between Original and Reconstructed Data



→ 28:1 → 18.7:1 → 14:1 → 11.2:1 → 9.33:1

Fig. 8 Fraction of Total Variance in each Eigenvalue



Fig. 10. Classified image - 4 codewords - compression ratio = 194.9.



Fig. 11. Classified image - 64 codewords - compression ratio = 25.1.

0-2

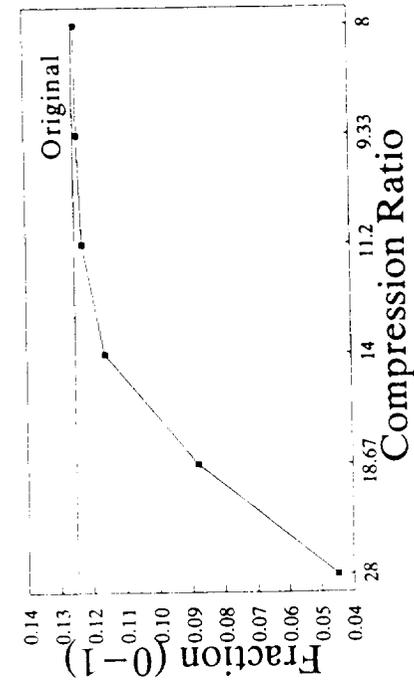


Fig. 9c Fraction of Total Variance in Eigenvalue No. 2

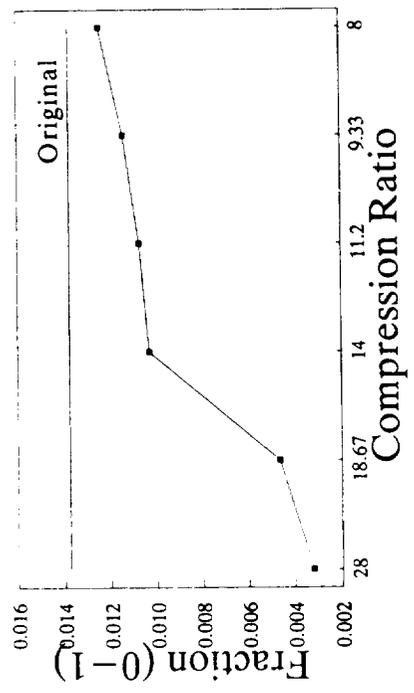


Fig. 9d Fraction of Total Variance in Eigenvalue No. 3

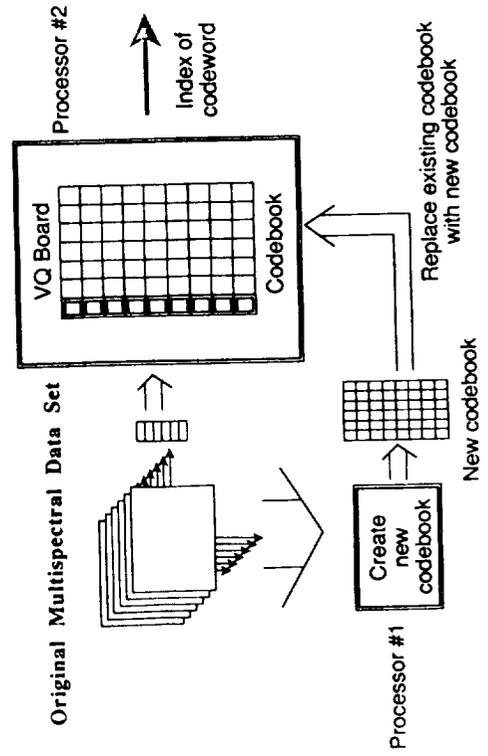


Fig. 12 Implementation layout for vector quantization

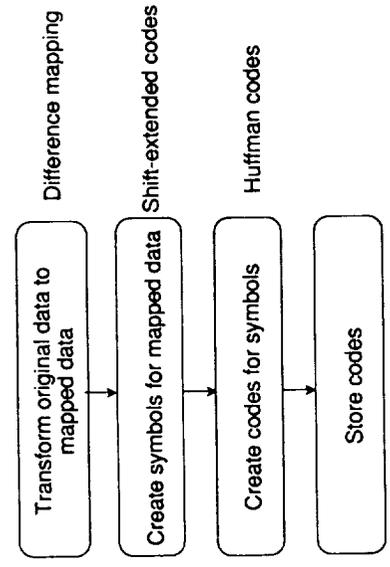


Fig. 13 Procedure for lossless coding.

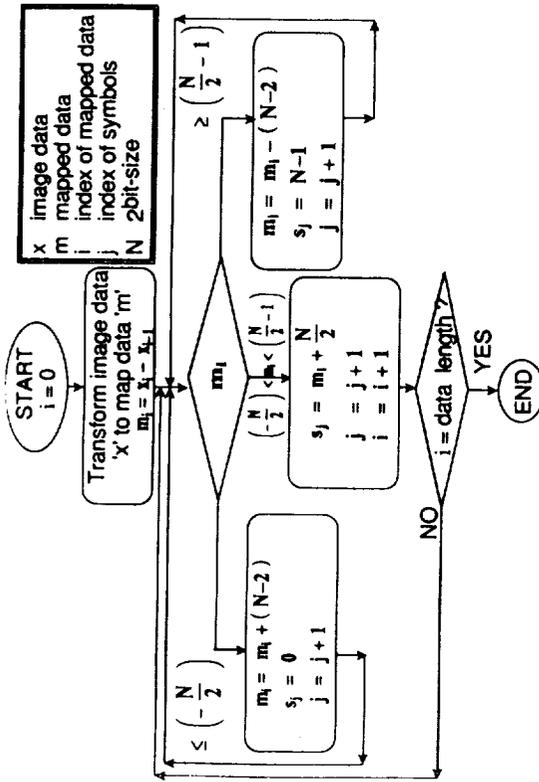


Fig. 14 Flow chart for difference mapping and shift extended coding

BITS	CODEWORDS							
	4	8	16	32	64	128		
2	6.67	5.41	3.85	1.99	1.60	1.31		
3	6.84	5.52	4.40	2.92	2.10	1.74		
4	6.96	5.84	4.44	3.51	2.51	2.15		
5	6.96	5.84	4.91	3.51	2.54	2.20		
6	6.96	5.84	4.91	3.86	2.67	2.22		
7	6.96	5.84	4.91	3.86	2.69	2.22		
8	6.96	5.84	4.91	3.86	2.69	2.22		

Fig. 15a Compression Ratios for lossless compression for CAMS images

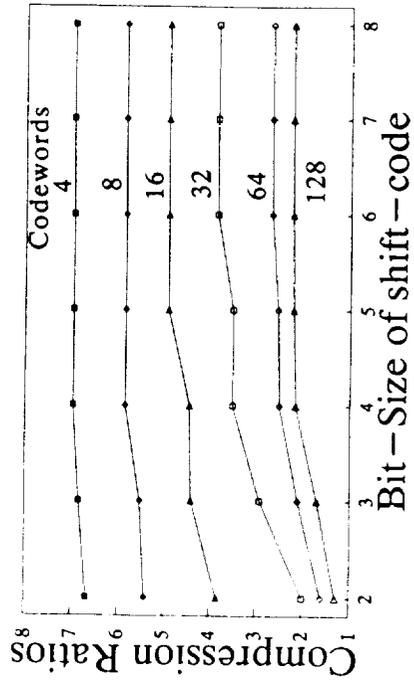


Fig. 15b Lossless coding on Vector Quantized images

VECTOR QUANTIZATION (LOSSY)	LOSSLESS			
CDWDS.	RATIO	ERRRATIO	RESULT.	
4	28.00	15.28	6.96	194.88
8	18.67	11.69	5.84	109.03
16	14.00	7.39	4.91	68.74
32	11.20	5.67	3.86	43.24
64	9.33	4.29	2.69	25.10
128	8.00	3.63	2.22	17.76

Fig. 16 Resultant compression ratios for CAMS images